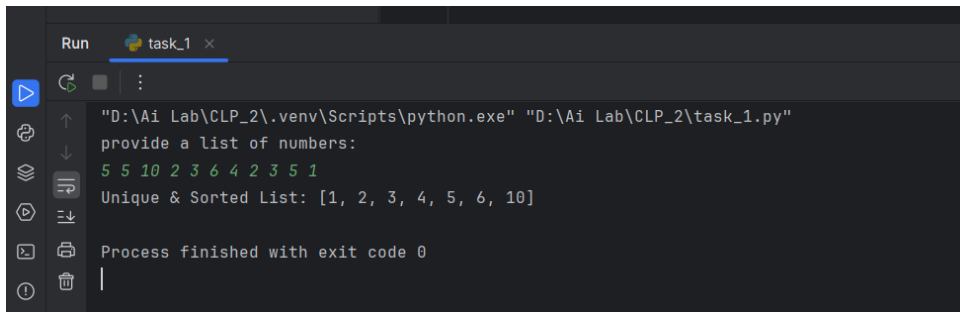


**Task 1:** Given a list of numbers, remove duplicates and sort in ascending order


```
1
2 print('provide a list of numbers:')
3 initial_list=list(map(int,input().split()))
4 unique_list=list(set(initial_list))
5 unique_list.sort()
6
7 print('Unique & Sorted List:',unique_list)
```



```
Run task_1 x
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_1.py"
provide a list of numbers:
5 5 10 2 3 6 4 2 3 5 1
Unique & Sorted List: [1, 2, 3, 4, 5, 6, 10]
Process finished with exit code 0
```

**Task 2:** Find the common elements between two lists using sets

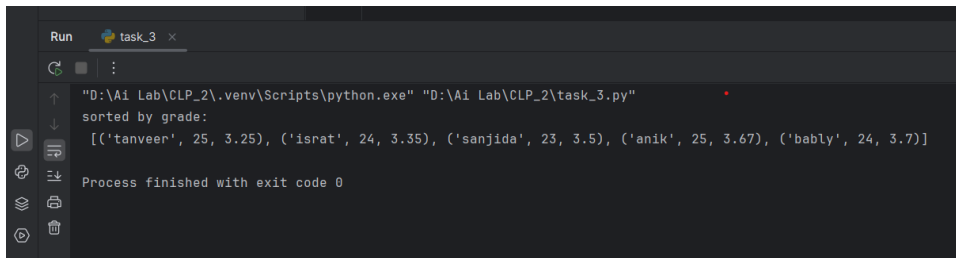
```
1
2 list_a=["apple", "banana", "cherry",5,7,10]
3 list_b=["google", "microsoft", "apple",5,12,10]
4
5 print('Input is:',list_a,'\nInput is:',list_b)
6
7 set_a=set(list_a)
8 set_b=set(list_b)
9
10 set_c=set_a.intersection(set_b)
11 print('common element is:',set_c)
```



```
Run task_2 x
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_2.py"
Input is: ['apple', 'banana', 'cherry', 5, 7, 10]
Input is: ['google', 'microsoft', 'apple', 5, 12, 10]
common element is: {10, 'apple', 5}
Process finished with exit code 0
```

**Task 3:** Create a tuple of student records (name, age, grade) and sort by grade

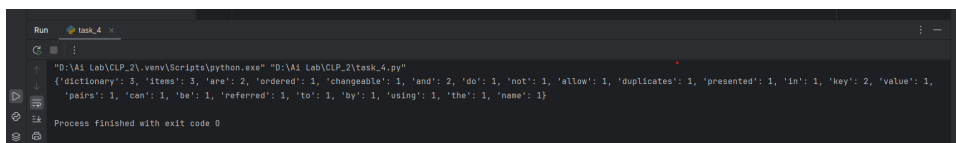
```
1 students=(
2     ("sanjida",23,3.5),
3     ("bably",24,3.7),
4     ("tanveer",25,3.25),
5     ("israt",24,3.35),
6     ("anik",25,3.67)
7 )
8
9
10 sorted_students = sorted(students, key=lambda student:student[2])
11 print('sorted by grade:\n',sorted_students)
```



```
Run task_3
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_3.py"
sorted by grade:
[('tanveer', 25, 3.25), ('israt', 24, 3.35), ('sanjida', 23, 3.5), ('anik', 25, 3.67), ('bably', 24, 3.7)]
Process finished with exit code 0
```

**Task 4:** Count word occurrences in a given text and store them in a dictionary

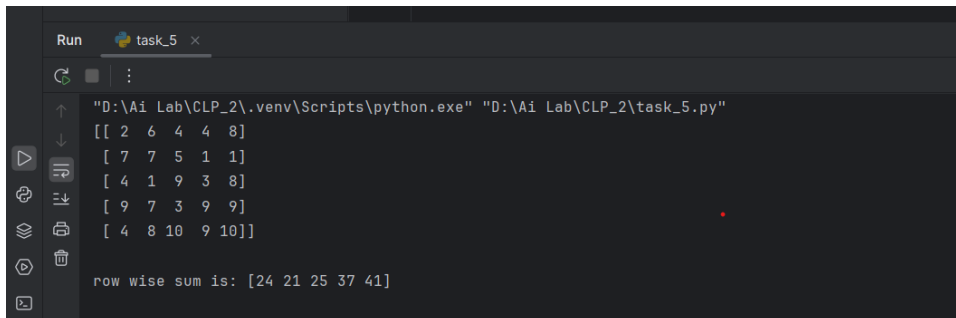
```
1
2 import pprint
3 import string
4
5 text="Dictionary Items Dictionary items are ordered, changeable, and do not allow
6     duplicates. Dictionary items are presented in key: value pairs, and can be
7     referred to by using the key name."
8
9 text=text.lower()
10 text=text.translate(str.maketrans('','', string.punctuation))
11 words=text.split()
12
13 dictionary={}
14 for word in words:
15     if word in dictionary:
16         dictionary[word]+=1
17     else:
18         dictionary[word]=1
19 print(dictionary)
```



```
Run task_4
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_4.py"
{'dictionary': 3, 'items': 3, 'are': 2, 'ordered': 1, 'changeable': 1, 'and': 2, 'do': 1, 'not': 1, 'allow': 1, 'duplicates': 1, 'presented': 1, 'in': 1, 'key': 2, 'value': 1, 'pairs': 1, 'can': 1, 'be': 1, 'referred': 1, 'to': 1, 'by': 1, 'using': 1, 'the': 1, 'name': 1}
Process finished with exit code 0
```

## Task 5:Generate a 5x5 matrix of random integers and compute row-wise sums

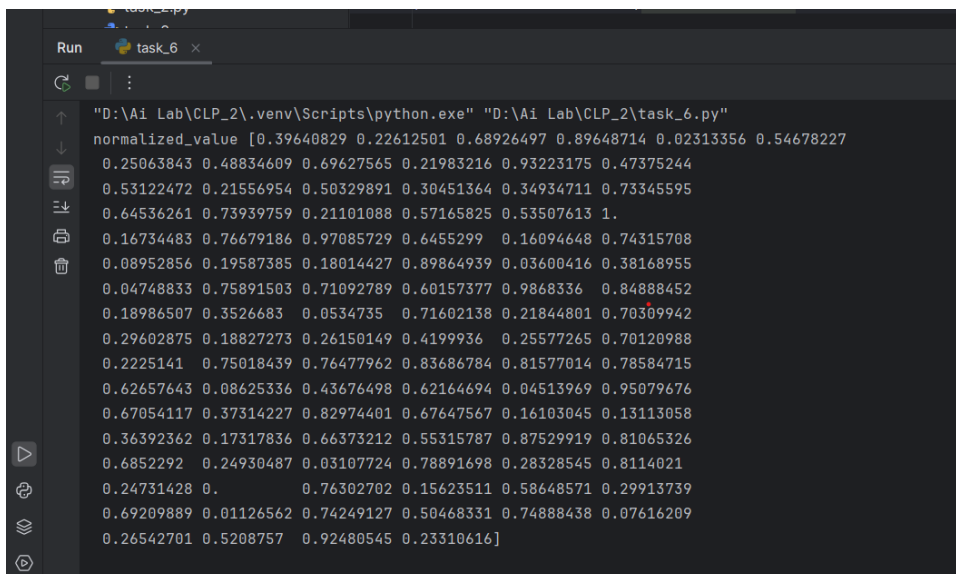
```
1
2 import numpy as np
3
4 matrix=np.random.randint(1,11 ,size=(5,5))
5 sum=np.sum(matrix , axis=1)
6
7 print(matrix,"\n")
8 print("row wise sum is:",sum)
```



```
Run task_5 x
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_5.py"
[[ 2  6  4  4  8]
 [ 7  7  5  1  1]
 [ 4  1  9  3  8]
 [ 9  7  3  9  9]
 [ 4  8 10  9 10]]
row wise sum is: [24 21 25 37 41]
```

## Task 6:Create an array of 100 random values and normalize them between 0 and 1

```
1
2 import numpy as np
3
4 random_values = np.random.random(100)
5 normalized_values = (random_values - np.min(random_values)) / (np.max(
    random_values) - np.min(random_values))
6
7 print("normalized_value",normalized_values)
```



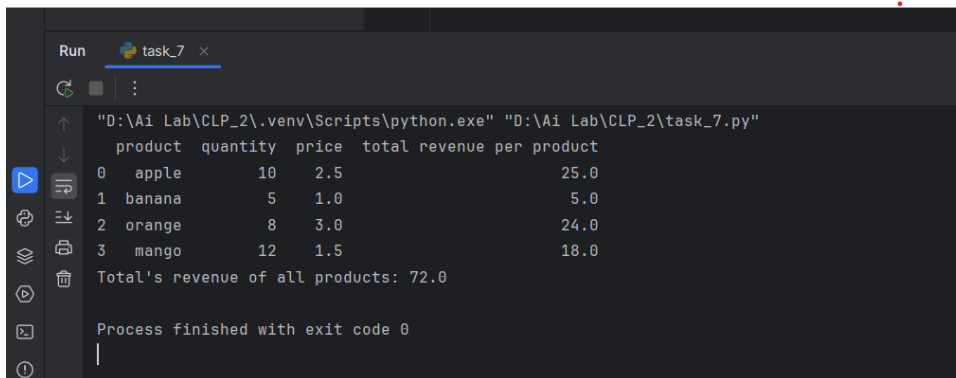
```
Run task_6 x
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_6.py"
normalized_value [0.39640829 0.22612501 0.68926497 0.89648714 0.02313356 0.54678227
0.25063843 0.48834609 0.69627565 0.21983216 0.93223175 0.47375244
0.53122472 0.21556954 0.50329891 0.30451364 0.34934711 0.73345595
0.64536261 0.73939759 0.21101088 0.57165825 0.53507613 1.
0.16734483 0.76679186 0.97085729 0.6455299 0.16094648 0.74315708
0.08952856 0.19587385 0.18014427 0.89864939 0.03600416 0.38168955
0.04748833 0.75891503 0.71092789 0.60157377 0.9868336 0.84888452
0.18986507 0.3526683 0.0534735 0.71602138 0.21844801 0.70309942
0.29602875 0.18827273 0.26150149 0.4199936 0.25577265 0.70120988
0.2225141 0.75018439 0.76477962 0.83686784 0.81577014 0.78584715
0.62657643 0.08625336 0.43676498 0.62164694 0.04513969 0.95079676
0.67054117 0.37314227 0.82974401 0.67647567 0.16103045 0.13113058
0.36392362 0.17317836 0.66373212 0.55315787 0.87529919 0.81065326
0.6852292 0.24930487 0.03107724 0.78891698 0.28328545 0.8114021
0.24731428 0. 0.76302702 0.15623511 0.58648571 0.29913739
0.69209889 0.01126562 0.74249127 0.50468331 0.74888438 0.07616209
0.26542701 0.5208757 0.92480545 0.23310616]
```

## Task 7: Load a CSV file of sales data and compute total revenue per product

```
1
2 import pandas as pd
3
4 data=pd.read_csv('files/sales_data.csv')
5 data['total revenue per product']=data['quantity']*data['price']
6 total_revenue=sum(data['total revenue per product'])
7
8 print(data)
9 print("Total's revenue of all products:",total_revenue)
```

CSV File Content (sales\_data.csv):

```
1 product,quantity,price
2 apple,10,2.5
3 banana,5,1
4 orange,8,3
5 mango,12,1.5
```



```
Run task_7 x
"D:\Ai Lab\CLP_2\.venv\Scripts\python.exe" "D:\Ai Lab\CLP_2\task_7.py"
product quantity price total revenue per product
0 apple 10 2.5 25.0
1 banana 5 1.0 5.0
2 orange 8 3.0 24.0
3 mango 12 1.5 18.0
Total's revenue of all products: 72.0
Process finished with exit code 0
```

## Task 8: Fill missing values in a dataset with column-wise means

```
1
2 import pandas as pd
3
4 data=pd.read_csv('files/dataset.csv')
5
6 print('Original Dataset with missing values:\n',data)
7
8 average_of_column=data.mean()
9 filled_data=data.fillna(average_of_column)
10 print("\n\nfilled dataset with columnwise mean value:\n",filled_data)
```

CSV File Content (dataset.csv):

```
1 col1,col2,col3,col4,col5
2 10,20,30,40,50
3 15,25,,45,55
4 5,35,45,,65
5 10,,50,60,70
6 20,30,,50,
```

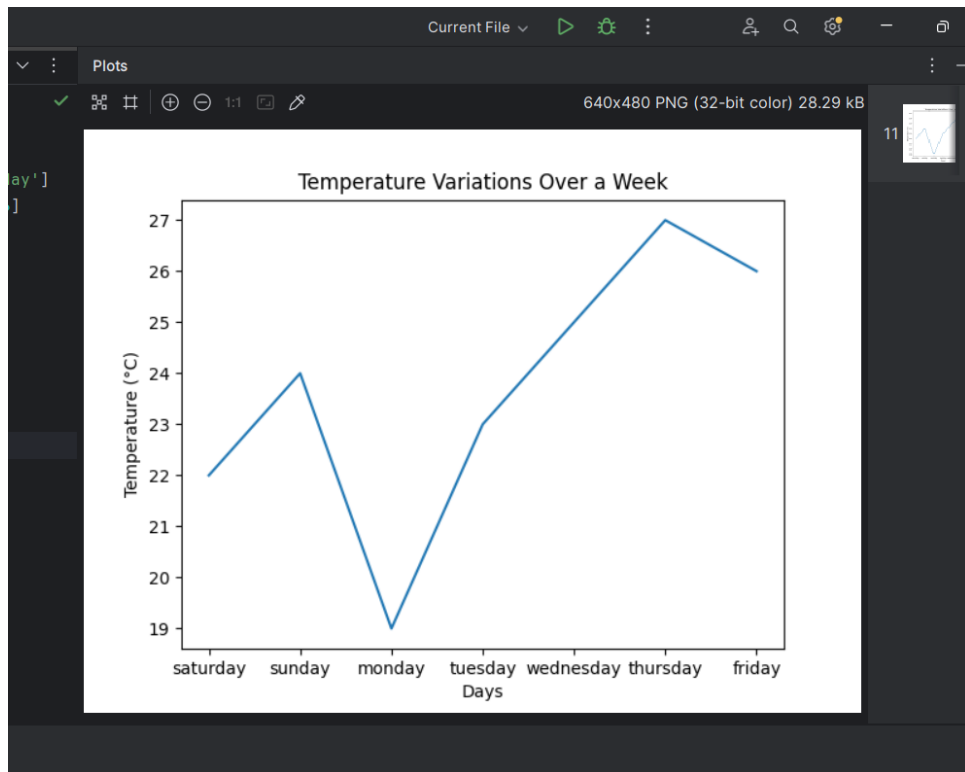
```
CLP_2 D:\AI Lab\CLP_2 0 print("\n\nfilled dataset with columnwise mean value:\n" f
task_8
Run
"D:\AI Lab\CLP_2\.venv\Scripts\python.exe" "D:\AI Lab\CLP_2\task_8.py"
Original Dataset with missing values:
  col1  col2  col3  col4  col5
0   10  20.0  30.0  40.0  50.0
1   15  25.0   NaN  45.0  55.0
2    5  35.0  45.0   NaN  65.0
3   10   NaN  50.0  60.0  70.0
4   20  30.0   NaN  50.0   NaN

filled dataset with columnwise mean value:
  col1  col2  col3  col4  col5
0   10  20.0  30.000000  40.00  50.0
1   15  25.0  41.666667  45.00  55.0
2    5  35.0  45.000000  48.75  65.0
3   10  27.5  50.000000  60.00  70.0
4   20  30.0  41.666667  50.00  60.0

Process finished with exit code 0
```

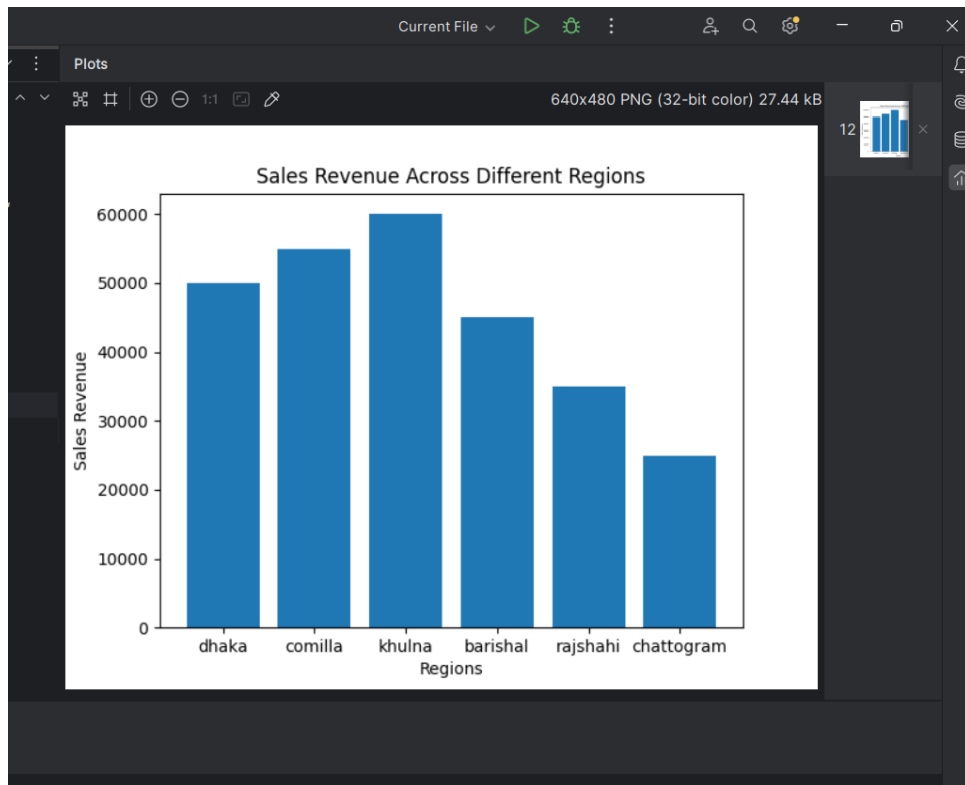
**Task 9:**Plot a line graph showing temperature variations over a week

```
1
2 import matplotlib.pyplot as plt
3
4 days = ['saturday', 'sunday', 'monday', 'tuesday', 'wednesday', 'thursday', '
    friday']
5 temperatures = [22, 24, 19, 23, 25, 27, 26]
6 plt.plot(days, temperatures)
7
8 plt.title('Temperature Variations Over a Week')
9 plt.xlabel('Days')
10 plt.ylabel('Temperature ( C )')
11
12 plt.show()
```



**Task 10:** Create a bar chart comparing sales revenue across different regions

```
1 import matplotlib.pyplot as plt
2
3
4 regions = ['dhaka', 'comilla', 'khulna', 'barishal', 'rajshahi', 'chattogram']
5 sales_revenue = [50000, 55000, 60000, 45000, 35000, 25000]
6
7 plt.bar(regions, sales_revenue)
8
9 plt.title('Sales Revenue Across Different Regions')
10 plt.xlabel('Regions')
11 plt.ylabel('Sales Revenue (bdt)')
12
13 plt.show()
```



## GitHub Repository

For more details: [https://github.com/sayed-2299/Academic/tree/main/Artificial%20Intelligence%20%20Lab\(CSE-316\)/CLP\\_2](https://github.com/sayed-2299/Academic/tree/main/Artificial%20Intelligence%20%20Lab(CSE-316)/CLP_2)